Student N

University of Saskatchev
Department of Computer Science
CMPT 117 Term 2 (Wayne Wang)
Feb 13, 2002

Mid-Term Examination

Time: 50 minutes   CLOSED BOOK   Total marks: 45

1. All questions on this examination sheet are to be answered. However, they can be answered in any order.
2. Read each question carefully and take time to plan your answer. A portion of the marks for each question will be awarded for the organization, clarity, and precision of the answer.
3. Apportion your time according to the indicated mark values.
4. All questions are to be answered in the space provided. Use the backs of the pages for rough work, indicating clearly that your answer is continued there.

Total Marks: 45

Question	Marks
1 (7 marks)	5
2 (15 marks)	9
3 (10 marks)	3
4 (13 marks)	8
Total (45 marks)	25

7   **1.. Short answers** (*1 mark each * 7 = 7 marks*)

(1) Give **two** advantages of Object-Oriented Programming over Structured Programming. Object-oriented programing allows for the reuse of code, it is also more easily debuged. X (0.5)

(2) What is the difference between a class and an object?

a class is used to define an object. The object usual contains some data. X (-1).

(3) What is the function of the *constructor* of a class?

The constructor is used to initialize the object an function

(4) When using an array of characters to represent a string, how can C++ tell the end of the string?

The end of a string ends with \0

(5) What are the **two** methods of passing parameters to a function?

- Pass by value
- Pass by reference.

(6) Give **one** advantage of *doubly linked lists* over *singly linked lists*.

doubly linked list is easier to move arround in. you can go in either direction.

(7) Define a new type **PINT** that is a pointer to an integer.

typedefine PinT = *int X (-0.5)

15   **2.. Read the following programs and write output for each of them (note that the line numbers are not part of the programs)**

6   (1) Pointers

```
01: #include <iostream.h>

02: void main(void)
03: {
04:    int *pa, *pb;
05:    int i, j;

06:    i = 3;
07:    j = 5;

08:    pa = &i;
09:    pb = &j;
10:    cout << "*pa=" << *pa << ",*pb=" << *pb << "\n";

11:    *pa = j;
12:    cout << "i=" << i << ",j=" << j << "\n";

13:    i = 7;
14:    pb = pa;
15:    j = *pb;
16:    cout << "i=" << i << ",j=" << j << "\n";
17: }
```

Start new line

After running the above program, the output is:

really output will line 10

*10: *pa ☑, *pb ☑*

12: i = 5̄ ✗, J = 5 ✓

16: i = 7 ✓, j = 5 ✗   (-2)

7

9   (2) Parameter passing

```
01: #include <iostream.h>

02: void swap1(int a, int b)
03: {
04:    int tmp = a;
05:    a = b;
06:    b = tmp;
07: }

08: void swap2(int* pa, int* pb)
09: {
10:    int tmp = *pa;
11:    *pa = *pb;
12:    *pb = tmp;
13: }

14: void swap3(int& a, int& b)
15: {
16:    int tmp = a;
17:    a = b;
18:    b = tmp;
19: }

20: void main(void)
21: {
22:    int x, y;

23:    x = 9;
24:    y = 11;
25:    swap1(x, y);
26:    cout << "x=" << x << ",y=" << y << "\n";

27:    x = 9;
28:    y = 11;
29:    swap2(x, y);
30:    cout << "x=" << x << ",y=" << y << "\n";

31:    x = 9;
32:    y = 11;
33:    swap3(x, y);
34:    cout << "x=" << x << ",y=" << y << "\n";
35: }
```

(a) The above program contains **one** *syntax error* inside the main function between lines *20 - 35*. The line number where the error exists is ___29___.

The correction to this line is ___swap1(x,y) ✗ (-2) swap2(&x, &y)___

(b) After this syntax error is fixed, the output of this program is:

x=9, y=11   x=9 y=11

x=9, y=9 ✓

x=11, y=9 ✓   (-2)

10   3.. Linked Lists

The following header file defines a *singly linked list* storing *integers*.

```
struct Node
{
    int item;
    Node* next;
};

class LinkedList
{
    protected:
        Node* head;
    public:
        LinkedList(){ head = NULL; };
        void InsertLast(int newItem);
}
```

The `InsertLast` member function inserts a new node to the end of the linked list and stores the `newItem` in this node. Please complete the body of this function below. **Hints**: it is possible that the linked list is empty before inserting the new node. In that case, the new node (with the `newItem` stored) becomes the **only** node of the list after the `InsertLast` is called.

```
void LinkedList::InsertLast(int newItem)
{
        if    (head->next = null)
        {      Node  n = new node;
               Node* nxT = n;
               head-> noxT = nxT;

                n -> noxT = null;
                n -> item = new Item;      2
                delete curs
        }
    else
        {   for (inT i = head -> noxT ; i != null ; i++)
            {  node[i] = cur ;      X

                Node n = new node;

                Cur -> noxT = n ;

                n -> noxT = null;

                n -> item = new Item ;

                delete cur;
            }
        }

    }
}
```

13   4. Programming

Write a program that reads in a sequence of non-zero *float* numbers from the keyboard, then outputs all the numbers that are greater than or equal to the average of the whole sequence. Note that the user indicates the end of the sequence by inputting **0**. However, this **0** does not belong to the sequence. **Hints**: you may use an *array* or a *linked list* to store the sequence. You can also use *STL* to simply the operations.

```
#include <iostream>
#include <fstream>
#include <vector>
using monospace std;

int main
{
    vector<float> entry new vector();
    float choice = 1;

    while (choice != 0) // when 0 is input program ends
    {
        cout << "enter next number\n";
        cin >> choice;
        entry.push back (choice); //adds choice to vector
    }

    float size;

    size = entry.size(); //members # of entrys in vector
    float x = 1
    for (int i=0; i < size; i++)
        { float j = entry.elementAT(i); //sets j = value at i
          float x = (*)j ; // multiplying all elements in entry
        }

    for(int i=0; i < size; i++)
        {
            if (entry.elementAT(i) >= X / size)      /07 -2
                cout << entry.element AT (i) ;
        }
}
```

(The End)